Nucleic Acids Research, 2020, Vol. 48, No. 11–15

关于 G4s 预测的计算方法指导

Emilia Puig Lombardi* and Arturo Londo˜no-Vallejo*

装

订

线

摘要

鸟嘌呤富集的核酸序列能够折叠形成非 B 形式的 DNA（non-B DNA）或者 RNA 结构，这种结构被称为 G4（G-quadruplexes）。最近，方法学的发展允许鉴定体外以及体内特定的 G4 结构，同时，计算机层面可以更高通量地预测这些 G4 结构，这大大拓宽了我们对 G4 关联功能的理解。传统上，保守的基序 $G_{3+}N_{1-7}G_{3+}N_{1-7}G_{3+}N_{1-7}G_{3+}$被用于鉴定一级序列上潜在的 G4 结构。此后，大量预测 DNA 或者 RNA 序列上 G4 形成潜力的算法发展起来了，同时也诞生出大量的跨物种全基因组层面的 G4 探索的研究。最近，新的方法学诞生了，这些算法需要额外考虑非典型序列以及结构倾向性结构稳定性。该篇综述旨在提供目前开源性的 G4 预测算法的最新概况，同时提供这些算法实现的直接样例。

关键词：G4s，正则表达匹配，滑动窗口，打分矩阵，机器学习，G4-seq

简介

G4 是由特定的 G 富集核酸序列形成的四链体二级结构。它们来自于大量稳定的‘G-四联体’的自我堆叠，在一个平面内，通常表现为四个由 Hoogsteen-type 氢键连接的鸟嘌呤的排列，这些鸟嘌呤的中心依靠一价的金属阳离子稳定结构（通常是 K+或者 Na+）（见图 1A）。大量的生理学研究和结构学研究揭示，G4 构象具有极其惊人的多样性，这取决于堆叠 G-四联体的数量，连接 loop 的长度与序列、核酸链在折叠过程中的取向以及中心离子通道中阳离子的性质。显然，G4s 能够通过分子内折叠，在一条单链 G 富集的 DNA 或者 RNA 上形成，或者在多条链上形成二聚体以及四聚体（图 1B）。大量证据表明，G4 序列具有多种必要的生物学功能，包括端粒的维持、DNA 复制、基因组的重排、DNA 损伤反应、染色体结构、RNA 加工以及转录翻译调控。尽管当前生物学相关的四链体的报道还主要关注在单分子折叠，但是与关键细胞功能有关的四链体分子间结构最近已经被发现。四链体结构的多样性，折叠的拓扑性以及体内的稳定性已经得到广泛地研究。因此，我们可以将其作为小分子（也可以称作 G4 配体）的一个新的药理靶点，进一步研究它的性质，这种小分子具备潜在的癌基因表达调控作用，或者能够发挥抗病毒活性。

目前，有许多广为流传的实验技术被用于验证特定序列形成 G4 的能力。这些技术包括提供结构信息的，比如 NMR（核磁共振）、X-ray 晶体学以及圆二色谱学，它们也常被用于监测四链体形成的动力学机制；提供热力学稳定性信息的，比如 UV melting（紫外线融化）；以及使用荧光标记可视化的。但是，这些方法都不能高通量地扫描全基因组并且识别新的 G4 结构。因此，

为了鉴定全基因组层面上推测的 G4 结构，必要的预测算法就产生了。事实上，迄今为止大多数描述 G4 结构的方法都结合了计算机预测和体外的生物物理学证据。有趣的是，第一个探测 G4 的计算算法是基于各种生物物理以及生物化学实验发展起来的。G4 检测的算法最开始使用正则表达式匹配的方法，后来，研究者通过打分计算结合滑动窗口进一步优化原本的模式匹配算法，直到最近又发展出机器学习的方法（表格 1）。

正则表达式匹配算法：传统算法

正则表达式是一段不连续的字符序列，它被用于定义某一种检索模式。这种方法基于严格的检测模式工作，我们假定形成 G4 的序列会具备这种模式。正如之前所提及的，生物物理学的数据定义了分子内 G4s 的基序序列，这种基序由有限长度的间隙分隔的鸟嘌呤组成，在接近生理的条件下，这种序列可以折叠成稳定的 G4s。Balasubramanian 以及 Neidle 小组的文献开创性地描述了对人类基因组 G4 形成潜力的第一次分析，其中鉴定出了 hg19 参考序列上 376, 000 个假定的单分子 G4s。他们使用正则表达式：$G_{3–5}N_{1–7}G_{3–5}N_{1–7}G_{3–5}N_{1–7}G_{3–5}$，这要求匹配的序列严格满足两个要求：其一是四个鸟嘌呤 run 都需要 3~5 bp 的长度；其二是，三个 loop 的长度应该在 1~7 bp 之间，环上核苷酸 N 可以代表{ A、T、C、G }中的任何一个核苷酸。许多脚本语言都提供了实现正则表达式匹配的框架，例如，下面的语法在 python 中使用来表示上述的正则表达式：'([gG]{3,5}\w{1,7}){3}[gG]{3,5}'（或者在 C 富集的链上使用的语法：'([cC]{3,5}\w{1,7}){3}[cC]{3,5}'，最后 G 富集以及 C 富集的模式都要被考虑进去）。这种搜索的形式会产生一个二元输出——"是或者否"，不需要对推断 G4 潜在的结构稳定性以及体内折叠的可能性做出任何判断。这种算法主要的难点在于评估嵌套结构，匹配找到的大量 G 堆叠长序列上有可能存在多个 G4 折叠，在这些区域界定单个的四链体可能比较模糊。我们提出如下两个简单的准则来处理重叠匹配：只计算不重叠的完全一致的基序；或者计算不同 loop 序列形式的重叠基序。例如，对于序列'GGGAGGGAGGGTGGGAGGG'，它可能得出两个 G4 基序，其中一个带有 A-A-T loop 但是另外一个带有 A-T-A loop。

从 2005 年以来，这种基序（或者使用不同长度的环限制相同底层表达的细微变体）被大多数研究所采用。直到最近，一项研究表述了使用相似的模式检索的方法识别人类基因组分子间 DNA 四链体，但是该研究同时考虑了两条 DNA 链。链交织 G4 的五种不同拓扑结构被提出，结构的不同取决于 G 或者 C 的顺序：（i）GGCC，（ii）GCGC，（iii）GCCG，（iv）GCCC，（v）CGCC。同时预测结果表明，这五种拓扑结构在人类参考基因组中广泛存在（hg19 参考基因组上存在 550, 977 条独特的链间 G4s 以及 374, 834 条链内 G4s 基序，这些 G4s 结构都允许环的大小在 1~7 bp）。一项最近的报告说明了利用这种分子间 G4 的预测方法，通过直接、基因组范围内的 DNA 双链断裂标记技术去评估酵母体内四链体形成和基因组稳定性之间的关系。

我们来看一下这个全局模式：$G_{x1}N_{L1}G_{x2}N_{L2}G_{x3}N_{L3}G_{x4}$，可以发现其中具备一些重要的特征——每个 loop 的序列长度 L 以及鸟嘌呤 run 的长度 x，这些特征能够用来区分和分类序列，以便后续的分析。考虑到环的长度在 1~7 bp 能够得出 21, 844 种可能的序列，然后结合三个环结构组合会产生 1, 013 种不同的模式，这就部分解释了为何关注核酸的 loop 识别的研究数量有限。Huppert 和 Balasubramanian 已经阐述过这个情况了，他们还提供了人类参考基因组上所有假设 G4 结构的 loop 长度的图谱。

滑动窗口以及打分方法

The quadruplex-forming G-rich sequence (QGRS) mapper 算法采用了和正则表达匹配算法稍微不同的策略：当出现多种可能性时，它通过对每个可能形成 G4 的序列打分，然后对这些序列进行排序，进而预测出最可能产生 G4 的序列。这个算法的实现采取了一种更加灵活的方式，它使用更宽泛的基序：$G_{x1}N_{L1}G_{x2}N_{L2}G_{x3}N_{L3}G_{x4}$，其中 G runs 的长度 x≥2 bp 而且允许最多一个间隙的长度为零。最终的打分策略则考虑三个方面：其一，短 loop 比长 loop 更常见；其二，loop 的大小相近；其三，三鸟嘌呤的数量越多，四链体越稳定。尽管如此，实验数据支撑形成的 G-score 产生的作用是有限的，而且在评分的过程中，往往缺少对某些属性的验证。

最近，实验证实，体内以及体外都存在不完整以及非典型的 G4。这种 G4 的序列缺少 4 个三鸟嘌呤 runs，因此它也不符合典型的 G4 基序模式。相应地，允许错误匹配（mismatch）、凸起（bulges）以及不完整四联体（G-triads）的新的 G4 预测工具应运而生。例如，ImGQfinder 在更加宽泛的鸟嘌呤 runs 长度的基础上，考虑了单个 bulge 或者 mismatch 的可能性；pqsfinder 则允许在单链中出现三种不完整的匹配（mismatch、bulges 以及长度大于 9 的长 loops）以及会对每个预测出来的 G4 进行打分，这使得序列和结构稳定性之间的关系能够被量化。尽管这种打分系统是经验主义近似的方式，pqsfinder 依旧为实验用户提供了比较灵活的预测框架，因为它允许用户自定义匹配和打分的标准。

另外一方面，基于滑动窗口策略的算法也发展了起来，渐渐被用于基因组上潜在 G4s 的侦测。这种探测的方法不需要定义严格的 PQS（predict quadruplex sequence）边界也不需要定义严格的序列组成成分，因此它能够识别非典型的 G4s，尽管代价是无法检测重叠的结构或者区域。The G-quadruplex potential（G4P）calculator 采用滑动窗口来评估鸟嘌呤 runs，滑动窗口策略取决于三个参数：每个鸟嘌呤 run 的长度（默认 k=3），滑动窗口的大小（默认 w=100）以及滑动的移动间距（默认 s=20）。在窗口大小为 w 时，算法从输入序列上用户定义好的起点出发，每次移动 s 个核苷酸，检索包含 4 个长度至少为 k 的鸟嘌呤 runs 的序列，最终 G4P 就产生了很多个包含四个 k-mer 鸟嘌呤的窗口，这些鸟嘌呤至少被 1 个核苷酸分隔。这种策略限制了 G4 候选序列的长度而不是这些序列内部 loop 的长度，这与体外发现长度大于 7 bp 的 loops 能够支持 G4 的形成是相吻合的。作者还公布了开源的在 Windows 系统上运行的程序，Ryvkin 和他的同事们还提供了 R 实现的伪代码。我们也稍加修饰并且展示如下（图 2）

有趣的是，最近，计算方法都引入了采取大量实验数据的验证阶段，这与之前提到的一些基于严格数量的生物物理实验归纳得到的方法不同。pqsfinder 测试了大量高通量的体内生成的 G4-seq 数据，这些数据被用来训练算法的参数。G4Hunter 算法使用了体外确认的 392 个 G4 sequence 以及人类线粒体基因组的 GC 富集区域的 G4 倾向的深入分析来测试和验证。这个工具通过计算 G 富集（G-richness，影响序列上鸟嘌呤片段）以及 G 偏斜（G-skewness，影响互补序列之间的 G/C 不对称性）用于获取 给定序列的 G4 倾向得分。引入这两个参数是考虑到 G4s 结构位点附近存在的胞嘧啶可能会导致实验扰动，因为胞嘧啶可以与鸟嘌呤互补配对并最终阻碍 G4 的形成。G4Hunter 采用 python 实现，需要设定两个参数，其一是窗口大小（默认设置为 25 bp），其二是分数阈值，用于检测 G4s 时选取分数符合的序列。作者表示，25 bp 的窗口适用于大多数

的验证分析，而且这似乎是一个明智的选择，因为这和大多数体外实验的 G4 的实际大小相吻合。对于分数阈值而言，选取 1.2 这个值可以很好地区分 G4 和非 G4 序列，而且很好地平衡了敏感度（sensitivity）和特异度（specificity）。设定一个更高的分数阈值（比如大于 1.7）确实能够减少假阳性率，但是也会导致过高的假阴性率。因此，为了更加详尽地探测基因组或者目标序列上潜在的 G4 结构，低阈值可能是更优的选择。这种方法的主要局限性在于 loop 碱基的打分前后是独立的（每一个 A 或者 T 的碱基得分等于 0 必然是不合理的，例如，带有单胸腺嘧啶的核苷酸显然比带有单鸟嘌呤的 loop 更加稳定）以及侦测的分数阈值是由用户凭借经验主义选取的。类似地，cG/cC 的打分方案是专门为 G4 RNA 设计的，它能够解决 G:C 碱基配对与 G 四链体组装需求的 Hoogsteen G:G 碱基配对的竞争问题。这种方法对存在 Cs 的目标序列进行罚分，通过计算两个不同得分（cG 以及 cC）之间的比率得到 Cs 对 G4 稳定性的负面效应，这两个得分和 G 或者 C runs 的数量成比例，而且对于长延伸会进行增量加权。该方法还采用实验验证（尽管只对两个标记了 20 个 G4 序列的相对较小的数据集进行了验证），于是产生了实验性的检测阈值：cG/cC 得分在 2.05 到 3.05 之间，这有利于稳定的 G4 形成，而且 cG/cC 得分越高，越有可能是 G4 折叠。然而，这些参数的设定似乎是任意的，因为检测的分数阈值以及公式的乘法因子都是基于没有被严格证明的启发式算法选取的。故而，目前 cG/cC 打分系统的实现不太容易支持基因组的检测，但是还是比较适合查询感兴趣的特定序列。

最近进展：机器学习方法

总的来说，之前的方法大多基于专家知识（生物物理学以及生物化学实验的数据，已经解决的结构中推断出的结论等），而且只考虑到了有限数量的已观测到的 G4 结构，不能完美描绘出 G4 多样性构象可能性的完整图像（已知或者未知）。如果纯粹通过计算研究预测新的构造或者序列，这些策略不一定合适。因此，该领域的最新方法将重心落在机器学习算法的发展上，这样就能让数据去驱动预测。这种方法避免了提前定义基序模式，减少了重复性假设，这使得预测非典型与意料之外的 PQS 的分析精度得到提升。但是，当需要它提供 G4 形成的预测特征的进一步信息时，机器学习的方法相对就比较晦涩（通常我们称之为“黑箱”）。G4RNA screener 的方法实现了一个最小的机器学习模型（单层前馈神经网络），通过识别有实验验证的 G4RNA 数据库内的 G4s 倾向性的序列（149 条 G4 以及 179 条非 G4 序列）进行自我训练，然后给出一个输入序列与已知 G4s 结构的相似性得分。这个模型被证明具有优秀的 RNA G4s 结构的预测能力，而且对于丢弃随机选择的转录本特别有效。该方法随后在含有体外检测到的约 4000 个 RNA G4s 的数据集（rG4-seq）上测试，相比较于 cG/cC 以及 G4Hunter 算法的分类性能，得出了相当的甚至更好的结果。G4RNA screener 最初是以命令行的形式发布的，但是有趣的是，大多数用户都不太熟悉这种实现形式，以至于作者最近又发布了能够方便工具使用的图形界面版本。最终，RNAfold 工具被集成到 Vienna RNA 二级结构预测软件包内，作为评估预测到的 rG4 序列折叠能的补充方法（表格 1）。

Quadron 算法采取相同的方式，使用树状梯度增加机器（tree-based gradient boosting machines, GBMs, 一个回归和分类算法）作为它的模型轴心框架。Quadron 采用一个在体外进行过广泛实验的 G4 数据集（超过 700000 条序列，该数据集来自于全基因组上的 G4-seq，尤其针对 DNA G4s）

训练，这个可以本地运行的图形界面程序会输出一个文件，其中包含探测到的 G4s 序列、序列的位点（起始位点，长度，终止位点）以及 G4s 序列在聚合酶停滞位点错配水平的预测得分。作者表示，分值高于 19 表示对应的 PQS 将会折叠成高度稳定的 G4 结构。但是，这个工具当前版本不能输出单个序列的得分（例如，当用户提供一条单链输入“GGGAGGGAGGGAGGG”）。该程序能评估典型序列基序的形成倾向（loop 大小最大为 12 bp），从而减少了传统模式匹配方法导致的假阳性率以及假发现率。然而，到目前为止，这些方法还没有扩展到非典型序列，这使得机器学习方法的优点被限制了。

在一组实验验证的数据上比较不同工具之间的表现差异

之前提到的软件（或者它们的改版）都是公开的、开源的，我们可以通过表格 2 中的链接访问到对应的 wed 页面或者 repositories 页面。特别需要说明的一点是，所有的独立程序都能在台式电脑的本地运行。在本次工作范围内，我们决定不提供或者测试那些非开源的预测工具。其中一个例子就是 Myong 实验室所描述的分析方法，使用线性回归和高斯回归模型预测 G4 折叠的潜能。即使这种方法在相关出版物中有详尽的描述，但是其原始代码是基于非开源的软件 MATLAB。

我们比较了不同的目前正在运作的开源性 G4s 预测工具在一个参考数据集上的表现。用于此次评估的参考 G4s 数据集包含了 392 条体外实验证实的 G4 序列，其中，298 条是阳性样本（即 G4），94 条是阴性样本（即非 G4）。我们必须申明这个序列数据集存在一些缺陷，因为它不太平衡（里面的 G4 阳性样本三倍于非 G4 的阴性样本，而且绝大部分序列的基序都是典型的）并且包含了大量的没有上下游序列的单序列。然而，这是目前唯一一个体外实验验证过的开源性 G4 序列数据集，因此我们选取它作为直接的工具表现标杆。为了计算表现矩阵（其中包括精确度，敏感度，特异度以及 Matthews 相关系数，见表 3），我们设置了评分阈值，从而使得每一项评分都会得到可能的最高取值，特别是我们采用表 4 中指定的参数来配置对应的工具。此外，对于那些有预测分数的工具，我们也比较 G4 和非 G4 序列的得分，然后测量 ROC 曲线下方区域的面积（AUC，见图 3B）。需要注意的是，G4Catchall 工具结合了序列打分和正则表达匹配，但是目前只输出了 G4Hunter 得分，因此不考虑它的 AUC 值计算。

正如表 4 所展示，没有一个工具能够识别所有的 G4 阳性序列，G4Hunter（宽松的得分阈值 0.7，仅仅是为了在这个有限的基准数据集上能够表现较好）和 QGRS Mapper（使用 web 工具给予的最宽松的参数）获得了最大的真阳性值。在这个小型数据集上，Quadron 表现与 Quadparser 相当，在我们看来这并不奇怪，因为在当前版本，这个机器学习模型只使用了 Quadparser 和 G4-seq 的输出进行训练，仅仅用于评估带有最大 12 bp loop 的典型序列基序的 G4 形成倾向。所有的打分算法得到的 G4 序列的平均得分都显著高于非 G4 序列的平均得分（pqsfinder: 63:6; G4Hunter: 1.64:0.15; QGRS: 65:35; G4 Grinder: 51:28; 见图 3A）。这个现象进一步验证了打分系统，这说明了在得分和体外 G4 结构的形成之间存在显著的关联。而且，当工具文档中没有提供特定的截断值时，它也提供了一个简单直观的参考值，可以作为区分序列的得分阈值（比如，所有 QGRS 中发现的得分低于 35 的序列都是非 G4 结构的序列）。最后，正如之前提到的，因为实验验证的 G4 数据集不太平衡，MCC 度量值是最中肯的性能评估参考。正像表格 4 中所看到的一样，当设定得分阈值为 25 时（尽管默认的截断值是 52），pqsfinder 优于其他所有算法（MCC=0.902）。最后，

当使用 Quadparser 评估初级序列时，采用伸长的 loop（1-12 bp，G4L1-12，MCC=0.635）执行检索会产生显著优于使用传统保守基序（1-7 bp，G4L1-7，MCC=0.543）执行检索的结果，这个发现和我们之前的经验估计一致。

在低复杂度序列上评估 G4 形成的潜力

正如前面所提到的一样，一些 DNA 和 RNA 序列可能具备形成多重叠 G4 结构的潜力，尤其是 G 富集的低复杂度位点（比如 CpG 岛，单二核苷酸、三核苷酸重复）以及 GC 富集的启动子区域。在这些特定的情形下，单个 G4 的界定变得难以确定，因为正则表达匹配算法在低复杂度区域通常无法考虑到 G4 的富集程度，同时滑动窗口的策略偏向于预测大量的潜在 G4s 而不能区分它们的单个边界。我们可以用一个具体的例子来解释这个问题：BCL-2 基因的启动子区域包含了 39 bp 的 GC 富集区域，该区域位于 P1 启动子上游，它能够同该序列上相同的核苷酸竞争形成互相排斥的重叠 G4 结构。从 hg38 参考基因组上提取出这段序列后，我们运行了多个不同的预测算法，结果表明，Quadparser 在使用默认设定下（正则表达匹配）鉴定该区域 G4 贫瘠，相反地，G4Hunter（滑动窗口）则给出了 15 倍于 Quadparser 的 G4s 潜力评估值。必须要提的是，python 和 R 实现的 G4Hunter 算法在处理重叠窗口时存在一些差异——R 语言的脚本鉴定 G4 结构时倾向于将所有可能的拓扑重叠序列整合到一条具备形成 G4 潜力的序列中而不是将它们分离；但是 python 脚本程序既会输出整合的窗口也会输出每一条重叠的基序。总而言之，这个算法是有价值的，它既能预测出所有重叠的单元，也能输出最终的预测结果以及它们的得分。QGRS Mapper 算法则能够预测典型的重叠基序，而且通过计算每个 G runs 的 Gs 数目以及 loop 长度给每一条输出赋予一个分值。类似地，pqsfinder 也能够通过参数调优最终得到序列内的所有重叠 G4s，同时提供覆盖输入序列每个位点的整体密度。当对 BCL-2 启动子区域的 GC 富集区域使用 pqsfinder 时，它可以鉴定出 23 条重叠的 G4 序列，其中 9 条带有较高的分值，被认为具有很高的 G4 折叠倾向。

在计算机以及体外针对 12 个物种进行 G4 序列质量的评估

最终，我们采用不同计算机预测方法（两种不同的方法，Quadparser 以及 G4Hunter 的 python 版本）比较了数据集（从最新版的 G4-seq 高通量测序方式获得）上的 G4 潜能。确实，12 个物种的全基因组 G4-seq 图谱最近公布，其中包括了不同大小和%GC 含量的基因组。我们从 GEO 仓库内检索编号 GSE110582 获取 BED 文件，其中对应每个物种中观测到的 G4s 结构位点。这些数据来自于两个实验条件，生理学 K+浓度以及额外添加 G4 稳定配体（pyrido statin，PDS）处理。重要地是，由于配体允许更多假定的 G4 结构被识别，PDS G4 稳定的条件下 G4-seq 表现出更显著的敏感度（12 个物种序列文库的平均敏感度从 31%上升到 66%）。另外，这个条件下平均特异度也增强了（从 81%上升到 85%），这可以解释为 PDS 处理条件下采用了更为严格的阈值，从而可能限制假阳性的数量。

两种方法推断 G4s 的总量总结在表格 5 内。在全基因组水平上，只要选取一个合理 G4Hunter 得分阈值，滑动窗口策略的 G4s 发现数量就系统性地高于传统正则表达匹配方法。但是，当设定更加严格的得分阈值（如 1.5）时，G4Hunter 发现的假定 G4 结构的数量就与 Quadparser 相当（hg19

参考基因组上发现了 646802 条 G4 序列）。类似地，当测序同时采用 K+和 PDS 处理的实验条件时，G4-seq 在体外发现的潜在 G4 数量也系统性地高于那些从简单序列中预测出来的数量，而且，这个值通常还与 G4Hunter 预测得到的 G4 数量值相似（表格 5）。

然后，为了评估不同预测方法的匹配度，我们分隔并且注释（通过基因组进行协调）了每个不同的数据集（图 5 以及附图 1，表格 5）。我们采用 HOMER 软件包展示基因组特征关联分析。简单来说，对于任何一个给定的基序，我们首先确定它到最近的转录起始位点（TSS）的距离并将这个基序分配给那个基因；然后，我们确定该基序中心区域的基因组注释信息，并指定注释类别。对于 12 个物种，我们观察到相似的注释类别而且在这 12 个物种的数据集之间还存在大量的重叠。然而，我们依旧使用 G4Hunter 预测了大量的假定 G4s（尤其是比较 G4L1-12 预测结果和 K+条件下的 G4-seq 数据时）。

的确，我们发现很多 G4 序列位于基因之间、LINE/SINE 以及重复区域（尤其是在人类、小鼠和斑马鱼的基因组上）。但是 G4Hunter 依旧保留与其他预测工具相似的预测比例，这表明很有可能是由低分辨率或者低复杂度区域内的高命中率所导致的。虽然如此，我们也观测到了 G4Hunter 预测结果和 K+加上 PDS 处理条件下的 G4-seq 结果存在更高的重叠，这可能暗示了大量的非典型 G4 结构被检测到了，而这种非典型 G4 结构无法被 Quadparser 检索出来而且其在 K+条件下会更不稳定。总之，根据人类基因组上 G4 折叠潜在位点的数量比广泛描述的约 375 000 个 PQS 和位点的图谱显著向上修正，这个结果支持我们的观点，这对之后研究非典型序列是极其重要的。

最后，我们特别针对只能通过 G4-seq 方法得到的序列进行了注释（和 G4Hunter 以及 Quadparser 都没有重复的序列），试图鉴定测序得到的伪结果。在这个分析中，我们决定使用最全面的注释文件，包括 hg19（人类），mm10（小鼠），danRer（斑马鱼）以及 dm6（果蝇）参考序列集。此外，我们还计算全基因组背景下的基因组特征富集程度，在此过程中，我们考虑到了参考基因组给定特征区域的总大小以及 G4s 基序和这个特征区域重叠的总大小（如图 6 以及附图 S2）。有趣的是，我们观测到简单重复序列位点处有假定的 G4 积累，而这些 G4 都没有出现在 G4Hunter 以及 G4-seq 鉴定出的位点上（如图 6）。另外，G4-seq 唯一鉴定出的 G4s 出现在特殊区域（如 3’以及 5’ UTR、启动子或者由高通量方法以及计算预测方法评估出来的容易富集 G4s 的基因组特征区域）的频次更低，这种现象可能和 G4-seq 方法的开发者提出的该方法的局限性有关，由于缺少覆盖度以及基因组 GC 富集区域的组装难题，导致了这些特殊区域鉴定单独 G4 的低分辨率。

总结讨论

所有的计算机 G4 预方法都有缺点和局限性，即便是最近这一领域大有进展同时引进了基于实验数据的验证阶段。对于第一组算法，基于正则表达匹配，由于多样性被严重限制了，导致了仅仅只有相同类别的结构能够被找到，这其中就有很多非典型 G4 结构（具有超过 4 个 G 束集以或者更长的 loop 的序列）被排除在外。对于机器学习的方法，目前的局限性主要是依赖于可获得的数据集的质量和数量：比如，G4 RNA 就只采用了由 149 个已经确定的 G4 结构序列组成的训练集进行训练，大多数最近的实验数据及都只能用于人类基因组，即便 Balasubramanian 课题组

最近公布了 12 个物种的 G4-seq 图谱。机器学习方法另外的一个难点则和用于 G4 潜力评估的参考基因组的质量有关，Ensembl 或者 UCSC Genomes 这些打数据库内的组装基因组都被精心挑选过，但是，大多数参考序列，重复序列的长 runs（小卫星序列，CpG 岛，低复杂度的单核苷酸或者双核苷酸重复序列）都遗失了或者被随意截断了，因为这些序列特别难被组装，这可能导致对全基因组层面的 PQS 含量的低估。最后，大多数本综述引用的预测工具都不能明确地用于解释由多个 G4 亚基形成的高阶序列。特别是，最近人类基因组上段粒序列的高阶组装引起了广泛的关注，这可能是探索新方向的主要焦点之一。目前，两种计算方法被用于预测高阶的或者是多聚体的 G4 结构——G4iM Grinder（也被用作第 i 个基序识别的工具）和 QPARSE（专门用于预测 G4 的高阶结构以及带有长 loop 或者发卡 loop 的 G4s）。

附图表：

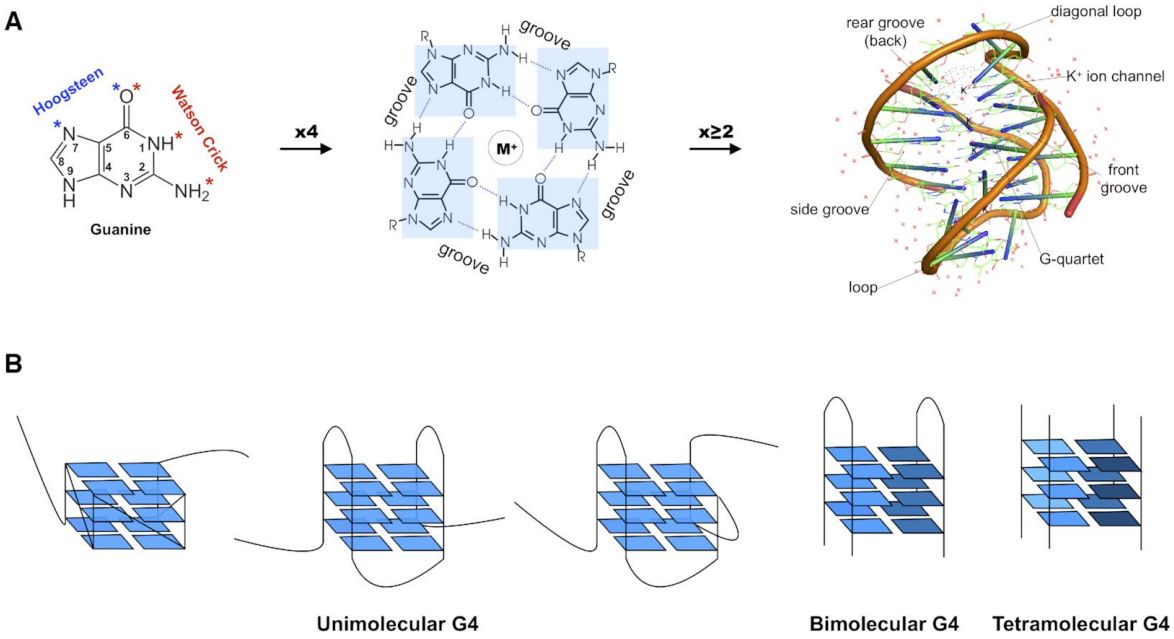


图 1 从鸟嘌呤到 G 四链体（G4）

(A) 从左到右：鸟嘌呤残基；由四个鸟嘌呤环绕、金属阳离子位于中心稳定的四平方面又称 G 四四方面；多个 G 四平方面的堆叠形成 G4 的二级结构。(B)多种多样的 G4 结构。从左到右：不同主链排列的三种单分子 G4s 构象（平行，反平行以及混合）；链与链之间的双分子 G4s 结构；链与链之间的四分子 G4 结构。蓝色的不同阴影代表了不同链。

```
#Generate random sequences to test
seqs <- paste(sample(c('A','C','G','T'), 1000000, prob = c(0.3,0.2,0.2,0.3), repl = T), collapse = '')
#G4 sequence patterns
gpat <- 'G{3,}.+?G{3,}.+?G{3,}.+?G{3,}'
cpat <- 'C{3,}.+?C{3,}.+?C{3,}.+?C{3,}'
#Parameters
w <- 100
s <- 20
n <- nchar(seqs)
t <- n/s
fwdcnt <- rep(0,t + 1)
revcnt <- rep(0,t + 1)
match.g <- NULL
match.c <- NULL
#Window match results
for (i in 0:t) {
        seqs.k <- substring(seqs, i*20+1, min(i*20+100,n))
        if(gregexpr(gpat, seqs.k, perl = T)[[1]][1]!= -1) {
                fwdcnt[i + 1] <- 1
                match.g <- c(match.g, regmatches(seqs.k,gregexpr(gpat, seqs.k, perl = T)))
        }
        if(gregexpr(cpat, seqs.k, perl = T)[[1]][1]!= -1) {
                revcnt[i + 1] <- 1
                match.c <- c(match.c, regmatches(seqs.k,gregexpr(cpat, seqs.k, perl = T)))
        }
}
#Final score calculation
g4p <- sum(fwdcnt)/length(fwdcnt)
c4p <- sum(revcnt)/length(revcnt)
#Matched sequences
match.g
match.c
```

图 2 R 实现的 G4P calculator 伪代码

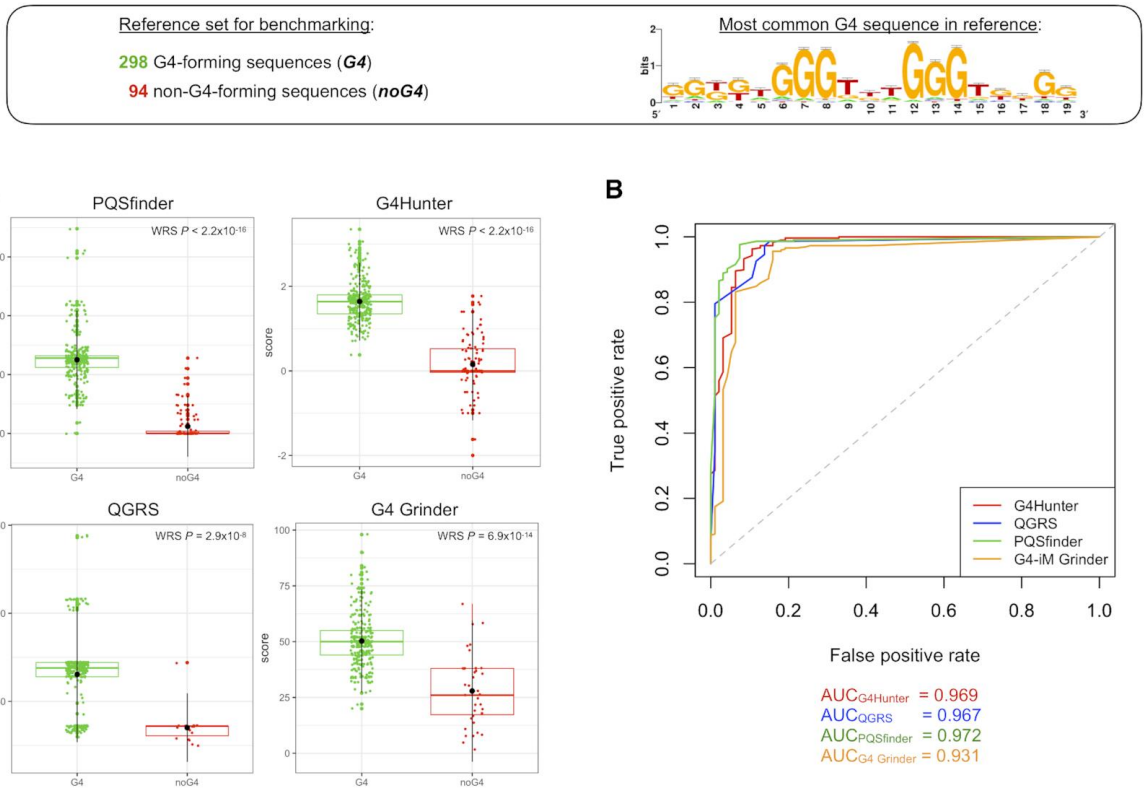
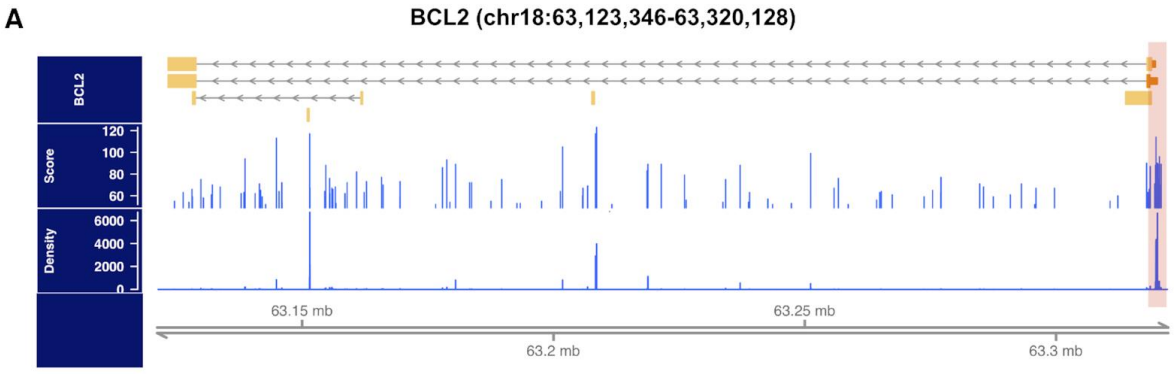
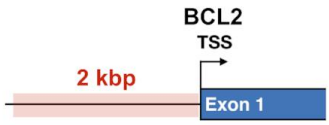


图 3 在一个参考数据集下比较不同 G4 预测工具的表现。用于评估的参考数据集包含 392 条体外证实的 G4 序列，其中 298 条是阳性样本，另外 94 条是阴性样本。序列的 logo 代表这个数据集中最可能发现的基序。(A) 不同工具对参考数据集的打分。点代表了每一条 G4 或者非 G4 序列的得分值。WRS：Wilcoxon rank sum test。(B) 参考数据集下不同工具得分的 ROC 曲线。一个随机评估的工具会表现为灰色对角虚线。每一个工具的 AUC 值展示在图片下方。

装 订 线



B



Tool	Settings	Predicted G4s
G4Hunter	All hits	33
	Score threshold: 1	19
Quadparser	Default (1 to 7nt loops)	2
	1 to 12nt loops	7
pqsfinder	Default	9
	Overlaps allowed	23

图 4 处理 GC 富集的基因启动子区域的重叠 G4 基序。

(A)轨道从上到下依次是：BLC2 基因注释（hg38 参考基因组上 18 号染色体的 123 346 到 63 320 128 位点）；所有 G4 基序 pqsfinder 预测得分的分布；所有 G4 基序 pqsfinder 预测密度的分布。密度值越高意味着区域的复杂度越低。(B)2kb，高密度的 BCL2 启动子区域的 G4 序列预测。表格展示了三个不同预测算法——Quadparser、G4Hunter 以及 pqsfinder 获得的结果。

装 订 线

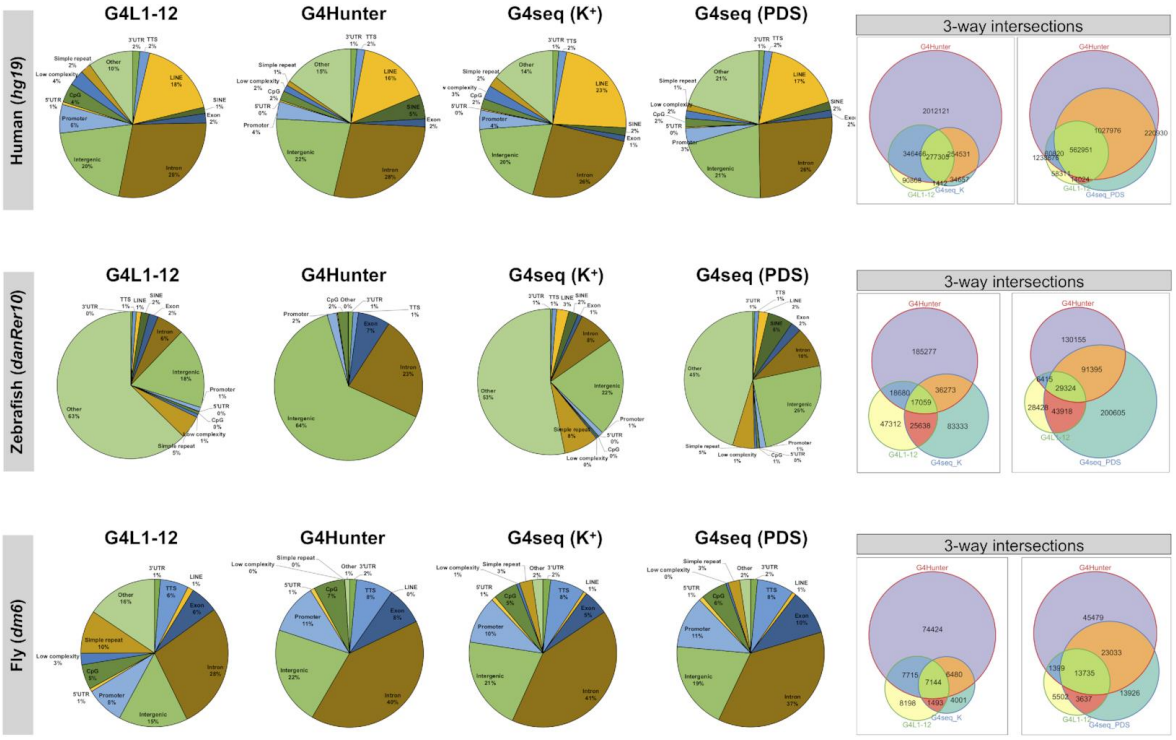


图 5 基因组上的不同预测方法发现的 G4 序列分布。三种不同方法（G4L1-12：正则表达匹配 G3-5N1-12G3-5N1-12G3-5N1-12G3-5；G4Hunter：滑动窗口和打分；G4-seq：体外高通量探测技术）预测得到的 G4 序列被注释。基因组上的特征分别从三个物种的注释文件中获得。不同数据集之间的三方重叠情况用加权 Venn 图表示（为清晰起见，采用面积比例的圆或者面）。

装 订 线

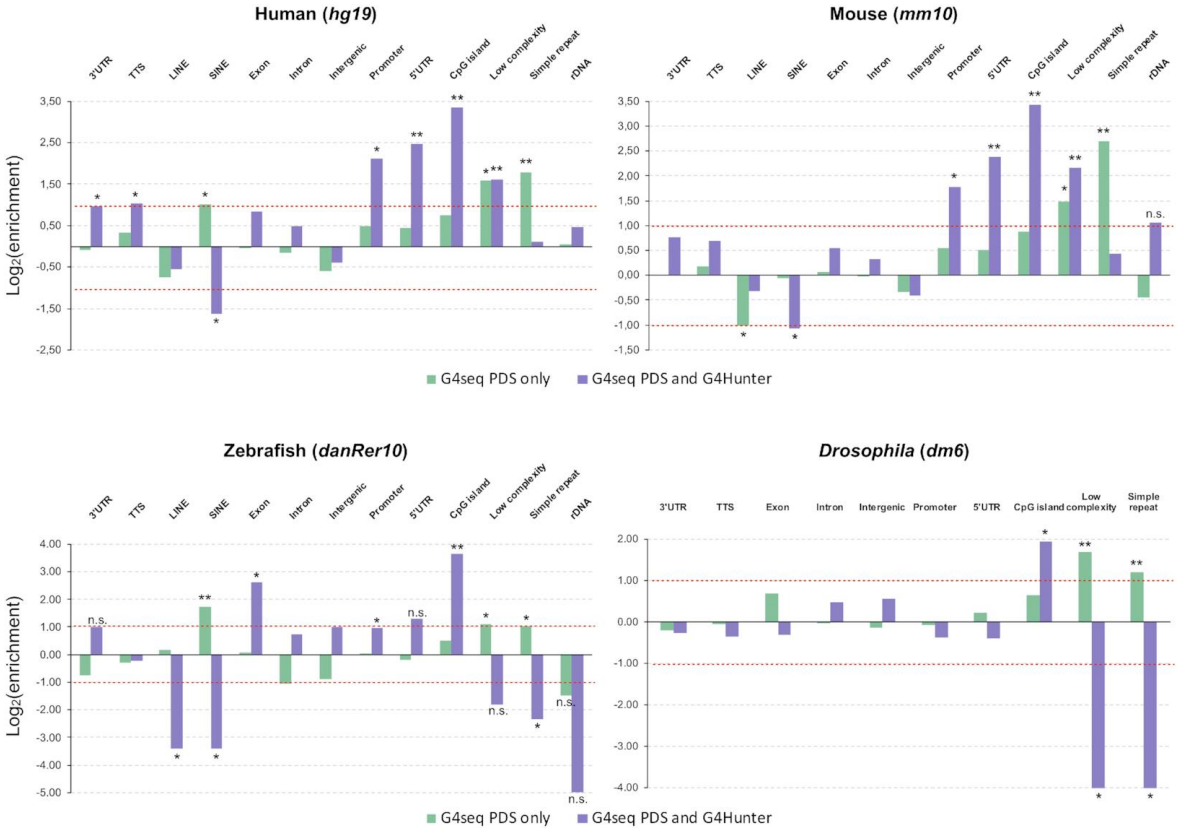


图 6 G4-seq 唯一识别的 G4 序列的注释。被 G4-seq 唯一发现的 G4 序列（绿色）的注释与同时被 G4Hunter 以及 G4-seq 发现的 G4 序列（紫色）进行比较。基因组特征从四个物种各自的注释文件中获得并且展示在 X 轴上。每一个进行比较的特征的 Log2(富集度)展示在 y 轴上。排列检验（n=100 的排列数）用于评估关联的重要性；**代表 P-value<0.01 以及|局部 z-score|>10；*代表 P-value<0.05 以及|局部 z-score|>10。

表格 1 开源的 G4 基序探测工具

Table 1.
Open-source G-quadruplex motif detection tools

Method	Name	Features	Language	Reference
Regular expression matching	Quadparser	$G_{t1}N_{L1}G_{t2}N_{L2}G_{t3}N_{L3}G_{t4}$, with t = 3–5 and L = 1–7 by default (G4L1–7)	C++, Python	Huppert and Balasubramanian (2005) (61)
	Quadruplexes	$G_{3-5}N_{1-7}G_{3-5}N_{1-7}G_{3-5}N_{1-7}G_{3-5}$	C++	Todd *et al.* (2005) (62)
	AllQuads	Intermolecular G4 detection:	Perl	Kudlicki (2016) (69)
		$G_{3+}N_{1-7}G_{3+}N_{1-7}C_{3+}N_{1-7}C_{3+}$		
		$G_{3+}N_{1-7}C_{3+}N_{1-7}G_{3+}N_{1-7}C_{3+}$		
		$G_{3+}N_{1-7}C_{3+}N_{1-7}C_{3+}N_{1-7}G_{3+}$		
		$G_{3+}N_{1-7}G_{3+}N_{1-7}G_{3+}N_{1-7}C_{3+}$		
		$G_{3+}N_{1-7}G_{3+}N_{1-7}C_{3+}N_{1-7}G_{3+}$		
	ImGQfinder	Allows to match imperfect intramolecular G4 sequences with a single defect: mismatches ($G_{k-1}NG_{k-1}$, where N = {A,T,C}, while a canonical G-run would be noted as G_k) or bulges ($G_{k-1}NG_{k-i+1}$, where N = {A,T,C})	Web	Varizhuk *et al.* (2017) (71)
Scoring	QGRS Mapper	$G_xN_{y1}G_xN_{y2}G_xN_{y3}G_x$, with x ≥ 2.	Standalone: Perl, Java; Web: PHP, Java	Kikin *et al.* (2006) (72)
		Restrictions: maximum length set to 30 nt, can be set to 45 nt by the user. A single loop of 0 length is allowed.		
		Scoring: G_{score}, benefits short and similar sized loops, and high number of tetrads; depends on the selected maximum G4 length.		
	pqsfinder	Three-step procedure:	C++ and R	Hon *et al.* (2017) (73)
		- step 1: identification of all possible G-run quartets;		
		- step 2: score assignment;		
		- step 3: overlap resolution		
Sliding window, scoring	G4P calculator	G-run length>3	C#	Eddy and Maizels (2006) (74)
		number of G-runs per window ≥4		
		window length 100 nt; and sliding interval length 20 nt		
	cG/cC	$\mathrm{cG(s)} = \sum_{i=1}^{n} (\lvert Gs(i)\rvert \times 10 \times i$	Spreadsheet treatment	Beaudoin *et al.* (2014) (75)
		$\mathrm{cC(s)} = \sum_{i=1}^{n} (\lvert Cs(i)\rvert \times 10 \times i$		
		$\mathrm{cG/cC\ score} = \mathrm{cG\ score/cC\ score}$		
	G4Hunter	Scoring based on G richness and G skewness: A,T: s = 0; G s > 0; C s < 0	R/Python	Bedrat *et al.* (2016) (76)
		Sliding window set at n = 25 nt by default		
		Window score = Sum(per-base values)/n		
		$\mathrm{Window\ score} = \sum_{i=1}^{25} s_i/n$		
Machine learning	G4RNA screener	Artificial neural network (ANN) trained with sequences of experimentally validated RNA G4s from the G4RNA database (77)	Python (*PyBrain* library)	Garant *et al.* (2017) (78)
	Quadron	Tree-based gradient boosting machine (GBM) algorithm trained on G4-seq data (79) for the human nuclear genome	R (*xgboost* library)	Sahakyan *et al.* (2017) (80)
Specialized tools	ViennaRNA folding suite (RNAfold)	Estimates RNA G4 (rG4) folding energy and assesses competition (minimum free energy comparison) between this fold and alternative RNA secondary structures (e.g. hairpin)	Web server Standalone: C	Lorenz *et al.* (2013) (81)
	G4PromFinder	Two-step procedure for the prediction of putative promoters in bacteria:	Python	Di Salvo *et al.* (2018) (82)
		- step 1: sliding window search over a query sequence (step: 1bp) until %AT reaches 40% ('AT-rich element');		
		- step 2: regular expression matching approach for G4 sequences, $G_xN_yG_xN_yG_xN_yG_x$, with 4 ≥ x ≥ 2, 10 ≥ y ≥ 1 and maximum length set to 30 nt		

表格 2 可获得的开源性 G4 探测软件

Table 2.
G-quadruplex detection open-source software availability

Name	Access	Author/maintainer	Implementation
AllQuads	http://moment.utmb.edu/allquads/	A. Kudlicki (69)	Perl
G4-iM Grinder	https://github.com/EfresBR/G4iMGrinder	E. Belmonte Reche (98)	R package
G4CatchAll	http://homes.ieu.edu.tr/odoluca/G4Catchall/	O. Doluca (99)	Web interface
G4Hunter	https://github.com/AnimaTardeb/G4Hunter	A. Bedrat (76)	Python
G4Hunter	http://bioinformatics.ibp.cz/	V. Brázda (100)	Web interface
G4Hunter	https://github.com/LacroixLaurent/	L. Lacroix (101)	R Shiny
G4P calculator	http://depts.washington.edu/maizels9/G4calc.php	J. Eddy (74)	Windows exe
G4PromFinder	https://github.com/MarcoDiSalvo90/G4PromFinder	M. Di Salvo (82)	Python
G4RNA screener	gitlabscottgroup.med.usherbrooke.ca/J-Michel/g4rna_screener	J.-M. Garant (78)	Python
G4RNA screener	http://scottgroup.med.usherbrooke.ca/G4RNA_screener/	J.-M. Garant (96)	Web interface
ImGQfinder	http://imgqfinder.niifhm.ru/	A. Varizhuk (71)	Web interface
pqsfinder	https://bioconductor.org/packages/release/bioc/html/pqsfinder.html	J. Hon (73)	R Bioconductor
pqsfinder	https://pqsfinder.fi.muni.cz/	J. Hon	Web interface
QGRS Mapper	http://bioinformatics.ramapo.edu/QGRS	O. Kikin, M. Viotti (72)	Web interface
Quadparser	https://github.com/dariober/	D. Beraldi	Python
Quadron	http://quadron.atgcdynamics.org/	A. Sahakyan (80)	R / R Shiny GUI

表格 3 用于评估工具表现能力的矩阵，能够直接从混淆矩阵计算获得

Table 3.
Performance metrics used for tool performance assessment than can be directly calculated from a confusion matrix

Metric	Use	Calculation
Sensitivity (SEN)	Measure the proportion of true positives (TP) that are correctly identified as such (true positive rate)	$\frac{TP}{TP+FN}$
Specificity (SPE)	Measure the proportion of true negatives (TN) that are correctly identified as such (true negative rate)	$\frac{TN}{TN+FP}$
False Discovery Rate (FDR)	Measure the proportion of false positives (FP) among positive results	$\frac{FP}{FP+TP}$
Accuracy (ACC)	Measure the proportion of true results (true positives and true negatives) among the total number of outcomes	$\frac{TP+TN}{TP+TN+FP+FN}$
Matthews Correlation Coefficient (MCC)	Discrete case for Pearson Correlation Coefficient; measures the quality of the binary classification by taking into account true and false positives (TP, FP) and true and false negatives (TN, FN)	$\frac{TP \times TN - FP \times FN}{\sqrt{(TP+FP)(TP+FN)(TN+FP)(TN+FN)}}$
Receiver Operating Characteristic (ROC) curve	Visualize the trade-offs between sensitivity and specificity when performing a binary classification	Plot the sensitivity values against the false positive rate (FPR, or 1 − SPE) at various thresholds (step: 0.1)
Area Under the ROC Curve (AUC)	Assess the probability that a true positive is scored greater than a true negative	Calculation based on trapezoidal rule

装 订 线

表格 4 G4 探测软件的表现能力比较

Table 4.
G-quadruplex detection software performance comparison

Tool	Settings	TP	FP	TN	FN	ACC	SEN	SPE	MCC	FDR
G4-iM Grinder	Default	233	1	93	65	0.832	0.782	0.989	0.671	0.004
G4-iM Grinder	Number of tetrads: 2; max loop len: 25; number of bulges: 3; score threshold: 31	285	16	78	13	0.926	0.956	0.830	0.795	0.053
G4CatchAll	Default	235	4	90	63	0.829	0.789	0.957	0.653	0.017
G4CatchAll	Min G-tract length: 2; max loop size: 15; max imperfections: 1; extreme loop allowed (for ≥3 G tracts)	293	20	74	5	0.936	0.983	0.787	0.820	0.064
G4Hunter (R scripts)	Score threshold: 1	278	7	87	20	0.931	0.933	0.926	0.823	0.025
G4Hunter (R scripts)	Score threshold: 0.70	294	15	79	4	0.952	0.987	0.840	0.864	0.049
ImGQfinder	Default (max loop length: 7; number of defects: 1), with number of tetrads: 3; max number of non-overlapping GQs	283	5	89	15	0.949	0.950	0.947	0.867	0.017
pqsfinder (R package)	Default	242	2	92	56	0.852	0.812	0.979	0.696	0.008
pqsfinder (R package)	Score threshold: 25	291	7	87	7	0.964	0.977	0.926	0.902	0.023
QGRS Mapper	Default	274	17	77	24	0.895	0.919	0.819	0.721	0.058
QGRS Mapper	Max length: 45; min G-group: 2; loop size: 0–36 nt, score threshold: 30	294	14	80	4	0.954	0.987	0.851	0.872	0.045
Quadparser (G4L1–7)	Default (7-nt loops): ([gG]{3, }\w{1,7}){3, }[gG]{3, }	196	2	92	102	0.735	0.658	0.979	0.543	0.010
Quadparser (G4L1–12)	12-nt loops: ([gG]{3, }\w{1,12}){3, }[gG]{3, }	225	2	92	73	0.809	0.755	0.979	0.635	0.009
Quadron	Default	225	2	92	73	0.809	0.755	0.979	0.635	0.009

TP: true positives; FP: false positives; TN: true negatives; FN: false negatives; ACC: accuracy; SEN: sensitivity; SPE: specificity; MCC: Matthews Correlation Coefficient; FDR: False Discovery Rate

表格 5 计算机发现的潜在 G4 以及 G4-seq 发现的体外潜在 G4

Table 5.
Potential quadruplexes found *in silico* and *in vitro* through G4-seq

								Annotation (G4Hunter ∩ G4seq K^+ + PDS, %)[a]								
Species	**Reference**	**Size (Mb)**	**%GC**	**G4L1–12**	**G4Hunter**	**G4seq K^+**	**G4seq K^+ + PDS**	**3'UTR**	**5'UTR**	**TTS**	**Exon**	**Intron**	**Intergenic**	**Promoter**	**Repeats[b]**	**LINE/SINE**
Human	*hg19*	3095.69	37.8	722 226	2 890 423	434 272	1 376 425	1.4	0.5	2.0	2.1	29.6	21.7	4.7	5.4	17.8
Mouse	*mm10*	2730.87	42.6	786 453	2 724 011	797 789	1 746 863	1.2	0.4	1.6	1.8	27.2	23.2	3.6	7.4	19.5
Zebrafish	*danRer10*	1371.72	36.8	103 252	263 185	141 637	321 230	1.0	0.2	1.0	9.2	21.9	61.7	2.4	2.5	0.0
D. melanogaster	*dm6*	143.73	42.1	24 804	110 024	19 399	55 263	1.4	0.8	7.7	9.1	39.4	21.6	10.7	8.2	NA
C. elegans	*ce11*	100.29	35.4	4290	36 136	4144	10 776	NA	NA	17.2	6.0	12.2	10.5	37.3	3.8	NA
S. cerevisiae	*sacCer3*	12.16	38.4	143	2 701	103	502	NA	NA	12.2	5.6	0.0	16.1	66.0	0.0	NA
L. major	*LmjFv6.1*	32.86	59.6	16 988	100 569	17 343	36 941	NA	NA	20.0	10.5	0.0	20.4	34.1	NA	NA
T. brucei	*Tb927*	35.83	46.8	3231	29 219	3 236	10 666	NA	NA	16.8	31.5	NA	7.5	42.4	NA	NA
P. falciparum	*Pfalciparum3D7*	23.33	19.6	193	4341	173	326	NA	NA	4.1	11.1	0.5	76.3	8.1	NA	NA
A. thaliana	*TAIR10*	119.67	36.1	2 849	25 786	2 407	11 953	NA	NA	16.7	42.3	2.2	11.9	26.2	NA	NA
R. sphaeroides	*ASM1290v2*	4.64	68.8	1 990	10 107	47	2291	NA	NA	19.8	3.9	NA	0.1	76.2	NA	NA
E. coli	*ASM584v2*	4.6	50.8	131	1701	291	5660	NA	NA	23.6	1.4	NA	NA	75.0	NA	NA

[a]Putative G4 sequences found by both G4Hunter and G4seq were annotated, results are shown as percentage (%) of motifs found in each category (NA, not applicable, indicates that a feature is not present in the annotation for the reference genome);
[b]The '*Repeats*' category includes regions annotated as low-complexity, simple repeats and CpG islands.

参考文献：见原文